## **Library Management System Documentation**

### **1. Project Overview**

The Library Management System is designed to manage various library operations, including item management, author management, patron management, and the borrowing and returning of items. The system provides a user-friendly interface to perform these tasks efficiently.

### **2. Class Descriptions**

#### **Library.java**

* **Description**: Manages the collection of publications and interactions with patrons.
* **Methods**:
  + +addPublication(pub: Publication): void - Adds a new publication to the library.
  + +getPublication(title: String): Publication - Retrieves a publication by its title.
  + +takeOutPublication(pub: Publication, patron: Patron): void - Allows a patron to borrow a publication.
  + +returnPublication(pub: Publication): void - Handles the return of a borrowed publication.

#### **LibraryMenu.java**

* **Description**: Provides a user interface for library operations.
* **Methods**:
  + +displayMainMenu(): void - Displays the main menu for library operations.
  + +borrowItem(): void - Initiates the borrowing process for a library item.
  + +returnItem(): void - Initiates the return process for a library item.
  + +addNewPublication(): void - Adds a new publication to the library.

#### **Author.java**

* **Description**: Represents an author with relevant details.
* **Attributes**:
  + -authorID: int - Unique identifier for the author.
  + -name: String - Name of the author.
* **Methods**:
  + +getAuthorID(): int - Returns the author's ID.
  + +getName(): String - Returns the author's name.
  + +setName(name: String): void - Sets the author's name.

#### **Book.java**

* **Description**: Represents a book in the library.
* **Attributes**:
  + -ISBN: String - International Standard Book Number.
  + -title: String - Title of the book.
  + -author: Author - Author of the book.
* **Methods**:
  + +getISBN(): String - Returns the ISBN of the book.
  + +getTitle(): String - Returns the title of the book.
  + +getAuthor(): Author - Returns the author of the book.
  + +setTitle(title: String): void - Sets the title of the book.

#### **BookAudio.java**

* **Description**: Represents an audio book.
* **Attributes**:
  + -narrator: String - Narrator of the audio book.
* **Methods**:
  + +getNarrator(): String - Returns the narrator of the audio book.
  + +setNarrator(narrator: String): void - Sets the narrator of the audio book.

#### **BookElectronic.java**

* **Description**: Represents an electronic book.
* **Attributes**:
  + -fileSize: int - File size of the electronic book.
* **Methods**:
  + +getFileSize(): int - Returns the file size of the electronic book.
  + +setFileSize(fileSize: int): void - Sets the file size of the electronic book.

#### **BookPrinted.java**

* **Description**: Represents a printed book.
* **Attributes**:
  + -pageCount: int - Number of pages in the printed book.
* **Methods**:
  + +getPageCount(): int - Returns the number of pages in the printed book.
  + +setPageCount(pageCount: int): void - Sets the number of pages in the printed book.

#### **Borrowable (interface)**

* **Description**: Interface for borrowable items.
* **Methods**:
  + +borrow(): void - Method to borrow an item.
  + +returnItem(): void - Method to return a borrowed item.

#### **Periodical.java**

* **Description**: Represents a periodical publication.
* **Attributes**:
  + -issueNumber: int - Issue number of the periodical.
  + -publicationDate: String - Publication date of the periodical.
* **Methods**:
  + +getIssueNumber(): int - Returns the issue number of the periodical.
  + +getPublicationDate(): String - Returns the publication date of the periodical.
  + +setPublicationDate(publicationDate: String): void - Sets the publication date of the periodical.

#### **PeriodicalElectronic.java**

* **Description**: Represents an electronic periodical.
* **Attributes**:
  + -fileSize: int - File size of the electronic periodical.
* **Methods**:
  + +getFileSize(): int - Returns the file size of the electronic periodical.
  + +setFileSize(fileSize: int): void - Sets the file size of the electronic periodical.

#### **PeriodicalPrinted.java**

* **Description**: Represents a printed periodical.
* **Attributes**:
  + -pageCount: int - Number of pages in the printed periodical.
* **Methods**:
  + +getPageCount(): int - Returns the number of pages in the printed periodical.
  + +setPageCount(pageCount: int): void - Sets the number of pages in the printed periodical.

#### **Publication (abstract class)**

* **Description**: Abstract class for publications.
* **Attributes**:
  + -publicationID: int - Unique identifier for the publication.
  + -title: String - Title of the publication.
* **Methods**:
  + +getPublicationID(): int - Returns the publication's ID.
  + +getTitle(): String - Returns the title of the publication.
  + +setTitle(title: String): void - Sets the title of the publication.

#### **Status (enum)**

* **Description**: Enumeration for publication status.
* **Values**:
  + AVAILABLE - Indicates the publication is available.
  + BORROWED - Indicates the publication is borrowed.

#### **Employee.java**

* **Description**: Represents a library employee.
* **Attributes**:
  + -employeeID: String - Unique identifier for the employee.
* **Methods**:
  + +getEmployeeID(): String - Returns the employee's ID.
  + +setEmployeeID(employeeID: String): void - Sets the employee's ID.

#### **Patron.java**

* **Description**: Represents a library patron.
* **Attributes**:
  + -patronID: int - Unique identifier for the patron.
  + -name: String - Name of the patron.
  + -address: String - Address of the patron.
  + -phone: String - Phone number of the patron.
* **Methods**:
  + +getPatronID(): int - Returns the patron's ID.
  + +getName(): String - Returns the patron's name.
  + +getAddress(): String - Returns the patron's address.
  + +getPhone(): String - Returns the patron's phone number.
  + +borrowItem(itemID: String): void - Allows the patron to borrow an item.
  + +returnItem(itemID: String): void - Allows the patron to return a borrowed item.

#### **Student.java**

* **Description**: Represents a student patron.
* **Attributes**:
  + -studentID: String - Unique identifier for the student.
  + -school: String - Name of the school the student attends.
* **Methods**:
  + +getStudentID(): String - Returns the student's ID.
  + +getSchool(): String - Returns the name of the student's school.
  + +setSchool(school: String): void - Sets the name of the student's school.

### **3. How to Start/Access the Application**

#### **Compiling the Project**

1. Use a Java compiler to compile all .java files.
   * Example command:

javac Library.java LibraryMenu.java Authors/Author.java LibraryItems/\*.java Patrons/\*.java

#### **Running the Application**

1. Run the LibraryMenu class to start the application.
   * Example command:

java LibraryMenu